



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**PROGRAM PRO ZOBRAZOVÁNÍ GENEALOGICKÝCH
DAT**

GENEALOGICAL PROGRAMME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

BOHUSLAV ŠTOUDEK

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Štoudek Bohuslav**

Obor: Informační technologie

Téma: **Program pro zobrazování genealogických dat
Genealogical Program**

Kategorie: Uživatelská rozhraní

Pokyny:

1. Nastudujte problematiku zpracovávání rodokmenů. Seznamte se s požadavky kladenými na genealogický software.
2. Navrhněte program, který umožní načítat data z GEDCOM souborů a vykreslovat jednotlivé rodokmeny podle zadaných parametrů. Dále musí umět vypisovat různé statistiky a histogramy (např. počet žijících obyvatel k určitému datu, histogramy věku při úmrtí pro zadané období, atd.)
3. Navržený program vytvořte.
4. Vytvořený program otestujte a navrhněte případná vylepšení do budoucna.

Literatura:

- Prostředníková Hana: Pokročilé zobrazování genealogických dat, bakalářská práce, Brno, FIT VUT v Brně, 2016.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Rozman Jaroslav, Ing., Ph.D.,** UITS FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Obsahem této práce je návrh a implementace programu pro zobrazování genealogických dat. Program zpracuje zdrojový soubor ve formátu GEDCOM a převede jej do grafické podoby. Dále zobrazuje rodokmeny a statistiky, ve kterých je možné filtrovat podle zadaných kritérií. Pro vývoj aplikace byly použity programovací jazyky C# a WPF.

Abstract

The content of this work is design and implementation of the program for displaying of a genealogical data. The program executes a source file in Gedcom format and converts it into a graphical form. Further it displays family trees and statistics in which it is possible to filter according to a chosen criteria. For the development of this program were used programming languages such as C# and WPF.

Klíčová slova

genealogie, GEDCOM, C#, WPF, XAML, MVVM, uživatelské rozhraní

Keywords

genealogy, GEDCOM, C#, WPF, XAML, MVVM, user interface

Citace

ŠTOUDEK, Bohuslav. *Program pro zobrazování genealogických dat*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Program pro zobrazování genealogických dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaroslava Rozmana Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Bohuslav Štoudek

17. května 2018

Poděkování

Tímto bych rád poděkoval Ing. Jaroslavu Rozmanovi, Ph.D. za odbornou pomoc při vytváření této práce. Dále bych rád poděkoval paní Miroslavě Zemanové za pomoc a poskytnuté informace.

Obsah

1	Úvod	2
2	Studium problematiky	3
2.1	Genealogie	3
2.2	Uživatelská rozhraní	7
2.3	Nástroje a technologie	7
3	Analýza požadavků a návrh řešení	10
3.1	Specifikace požadavků	10
3.2	Studium cílové skupiny a případy užití	10
3.3	Existující řešení	11
3.4	Návrh uživatelského rozhraní	13
3.5	Nástroje pro tvorbu GUI	15
4	Realizace	16
4.1	ViewModely	16
4.2	Zpracování dat	16
4.3	Vykreslování rodokmenů	17
4.4	Statistiky	18
4.5	Nastavení	19
4.6	Ukládání předchozích vybraných osob	20
4.7	Výsledné uživatelské rozhraní	20
5	Testování	24
5.1	Struktura dotazníku	24
5.2	Odpovědi respondentů	25
5.3	Další vývoj	28
6	Závěr	29
	Literatura	30
A	Obsah CD	32

Kapitola 1

Úvod

V poslední době stále roste zájem o informace o rodinném původu. Lidé se čím dál více zajímají o historii svého rodu, mnohé z nich to zaujme a stane se z toho jejich koníček, či dokonce práce. Data o původu jsou často ukládána v elektronické podobě a k jejich zobrazení a zpracování se používají genealogické programy.

Tato práce se zabývá návrhem a implementací genealogického programu, jehož hlavním úkolem bude zobrazování genealogických dat, která jsou uložena v elektronické podobě ve formátu GEDCOM. Program bude také zobrazovat všechny základní rodokmeny a statistiky ve formě grafů.

Začátek práce se zaměřuje na teoretické znalosti, které jsou nezbytné pro návrh a implementaci aplikace. V této části je vysvětlen pojem genealogie a popisují se zde jednotlivé druhy rodokmenů a způsoby získávání dat. Stručně jsou zde popsány nástroje a technologie, které bylo nutné nastudovat pro vývoj aplikace.

V další kapitole se práce zabývá analýzou požadavků a návrhem programu. Jsou zde popsány všechny cíle aplikace, studium cílové skupiny a existující řešení. Také je zde popsán návrh grafického uživatelského rozhraní.

V následující kapitole je rozebrána realizace aplikace. Jsou zde popsány postupy, které byly použity při implementaci programu. Dále je zde popsáno výsledné uživatelské rozhraní.

Poslední kapitola se zabývá testováním aplikace. Jsou zde rozebrány výsledky testování a navrženy postupy pro případně zlepšení. Dále je zde rozebrán další možný vývoj aplikace.

Kapitola 2

Studium problematiky

Tato kapitola se věnuje studiu teoretických informací, které byly nezbytné pro pochopení genealogické problematiky a pro tvorbu dalších částí této práce. Nejprve je zde vysvětlen pojem genealogie, jsou probrány 3 základní druhy rodokmenů a způsob získávání genealogických dat. V další části jsou stručně popsány technologie a nástroje potřebné pro vytvoření programu.

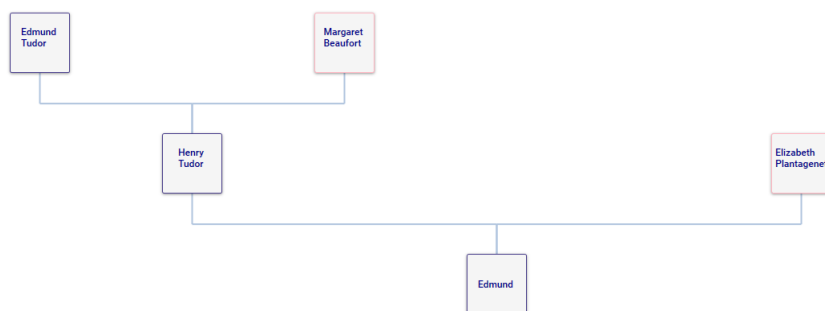
2.1 Genealogie

Genealogie

Genealogie je pomocná věda historická, která zkoumá vztahy mezi lidskými jedinci, vyplývající z jejich společného rodového původu. Genealogie se buď zabývá studiem jednotlivých osobností, nebo sledováním proměn jednotlivých druhů vztahů[3].

Rodokmen

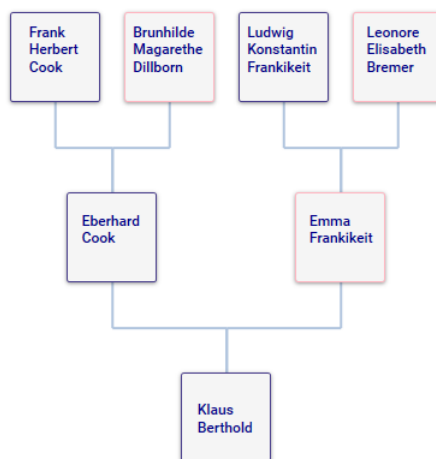
Rodokmen je nejjednodušší genealogická tabulka, která uvádí předky výchozí osoby pouze po otcovské linii. Sleduje tedy pouze mužské předky – otce, děda, praděda atd., přičemž o manželkách jsou uvedeny pouze základní informace. Typickým znakem rodokmenu je tedy společné příjmení. Jedná se o genealogickou tabulku, která poskytuje přehled pouze o části předků výchozí osoby, častěji se pro zobrazení rodinného stromu využívá vývod [7].



Obrázek 2.1: Ukázka rodokmenu

Vývod

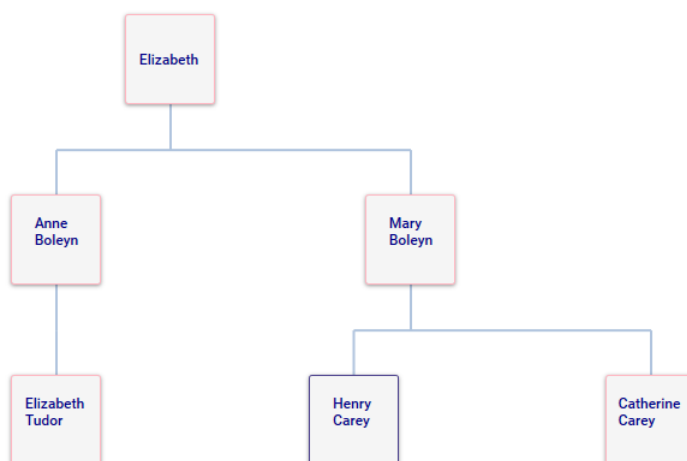
Vývod je nejpoužívanější formou genealogické tabulky. Zahrnuje všechny předky výchozí osoby. Při vytváření vývodu se postupuje zpět do minulosti po všech jeho liniích, a to jak mužských, tak ženských. Vývod je zobrazován v podobě stromu, kde výchozí osoba je jeho kořen a předci se rozvíjejí od kořene do koruny[12].



Obrázek 2.2: Ukázka vývodu

Rozrod

Rozrod je nejkomplikovanější formou genealogické tabulky. Obsahuje potomky výchozí osoby, tedy ty osoby, které mají společného předka. Existují dva typy rozrodu. Prvním je sledování jednoho příjmení, strom v takovém případě obsahuje pouze potomky synů nebo nemanželské potomky dcer. Druhou variantou je sledování potomků synů i dcer, bez ohledu na příjmení[8].



Obrázek 2.3: Ukázka rozrodu

Získávání dat

Genealogie vychází z historických pramenů, které jsou často v latinském nebo německém jazyce. Data se získávají z úředních a veřejných knih, listin a matrik. Povinné vedení matrik bylo zavedeno až v 18. století, proto se starší údaje dohledávají hůře. V současné době jsou již některé matriky převedeny do elektronické podoby a existují internetové stránky, na kterých se dají vyhledávat předci.

GEDCOM

GEDCOM je souborový formát pro výměnu genealogických dat mezi genealogickými programy. Označován je příponou .ged. Soubory GEDCOM jsou čistě textové a obsahují genealogické informace o individuálních osobách a data, která spojují tyto záznamy k sobě.

Soubor GEDCOM sestává ze záhlaví, sekce záznamů a koncového záznamu. Záznamy reprezentují individuální osoby, rodiny, zdroje informací a další rozmanité informace, včetně poznámek[2].

Jednotlivé záznamy jsou uloženy ve strukturovaném formátu po řádcích. Každý řádek začíná číslem úrovně, které indikuje úroveň zanoření. Hlavní záznamy(osoba,rodina,..) začínají číslem úrovně 0. Následující řádky patřící k hlavnímu záznamu mají číslo úrovně větší než 0. Za číslem úrovně může být křížový odkaz, který jednoznačně identifikuje záznam. Poté se na řádku nachází značka, která určuje význam řádku. Na posledním místě se může vyskytovat hodnota.

level + delim + [optional_xref_ID] + tag + [optional_line_value] + terminator

Obrázek 2.4: Struktura řádku GEDCOM souboru

Pro funkčnost aplikace jsou nejpodstatnější záznamy osoba a rodina. Záznam osoba obsahuje veškeré známé údaje o konkrétní osobě. Na obrázku 2.5 je zobrazena zjednodušená struktura záznamu o osobě s nejdůležitějšími informacemi. Konkrétně jednoznačný identifikátor záznamu, jméno, pohlaví, události a vlastnosti.

```
n @XREF:INDI@ INDI
+1 <<PERSONAL_NAME_STRUCTURE>>
+1 SEX <SEX_VALUE>
+1 <<INDIVIDUAL_EVENT_STRUCTURE>>
+1 <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>>
```

Obrázek 2.5: Zjednodušená struktura záznamu o osobě

Rodina popisuje vztahy mezi konkrétními osobami. Na obrázku 2.6 je zobrazena zjednodušená struktura záznamu o rodině s nejdůležitějšími informacemi. Konkrétně jednoznačný identifikátor záznamu, křížový odkaz na manžela a manželku, křížový odkaz na děti a události.

```
n @XREF:FAM@ FAM
+1 HUSB @<XREF:INDI>@
+1 WIFE @<XREF:INDI>@
+1 CHIL @<XREF:INDI>@
+1 <<FAMILY_EVENT_STRUCTURE|>>
```

Obrázek 2.6: Zjednodušená struktura záznamu o rodině

```
0 @FATHER@ INDI
1 NAME /Father/
1 SEX M
1 BIRT
2 PLAC birth place
2 DATE 1 JAN 1899
1 DEAT
2 PLAC death place
2 DATE 31 DEC 1990
1 FAMS @FAMILY@
0 @MOTHER@ INDI
1 NAME /Mother/
1 SEX F
1 BIRT
2 PLAC birth place
2 DATE 1 JAN 1899
1 DEAT
2 PLAC death place
2 DATE 31 DEC 1990
1 FAMS @FAMILY@
0 @CHILD@ INDI
1 NAME /Child/
1 BIRT
2 PLAC birth place
2 DATE 31 JUL 1950
1 DEAT
2 PLAC death place
2 DATE 29 FEB 2000
1 FAMC @FAMILY@
0 @FAMILY@ FAM
1 MARR
2 PLAC marriage place
2 DATE 1 APR 1950
1 HUSB @FATHER@
1 WIFE @MOTHER@
1 CHIL @CHILD@
0 TRLR
```

Obrázek 2.7: Ukázka GEDCOM záznamu[9]

2.2 Uživatelská rozhraní

Stroj nabízí funkce. Uživatel chce použít stroj ke svému užitku – chce využít jeho funkce. Cílem interakce je efektivní provoz a řízení stroje na straně uživatele, a zpětná vazba od stroje, což pomáhá uživateli při rozhodování dalších akcí.

Uživatelské rozhraní je souhrn způsobů, jakými lidé ovlivňují chování strojů, zařízení, počítačových programů či komplexních systémů [10].

Uživatelská rozhraní dělíme podle typu na:

- grafické uživatelské rozhraní
- textové uživatelské rozhraní
- příkazový řádek

Tato aplikace bude využívat grafické uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků. Na monitoru počítače jsou zobrazeny okna, ve kterých programy zobrazují svůj výstup. Uživatel používá klávesnici, myš a grafické vstupní prvky jako jsou menu, ikony, tlačítka, posuvníky, formuláře a další [4].

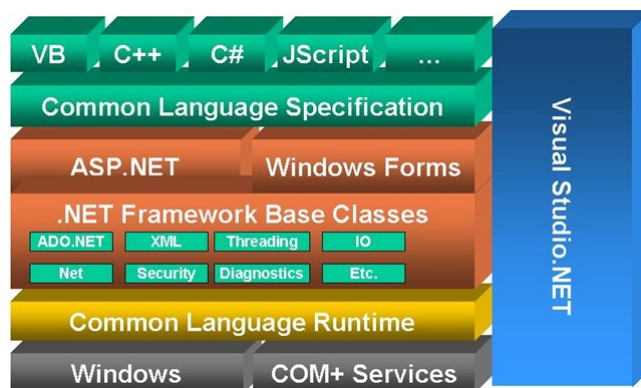
2.3 Nástroje a technologie

.NET Framework

.NET Framework je rozsáhlá softwarová platforma, která je určena pro vývoj mnoha různých druhů aplikací. Za pomoci .NET Frameworku můžeme vyvíjet nejen klasické aplikace pro Windows, ale mimo jiné i webové aplikace a služby, aplikace pro mobilní zařízení a mnoho dalších.

Celý framework obsahuje kromě velké sady knihoven a funkcí i samotné běhové prostředí, které zajišťuje běh a kompilaci aplikací. Obrovská škála funkcí nám zajišťuje to, že se nemusíme starat o psaní věcí, které se používají často. Díky běhovému prostředí pak budou naše aplikace rozumně rychlé a bezpečné. Vývoj v .NETu je rychlý, pohodlný a méně náchylný na chyby programátora.

V souvislosti s .NET Frameworkem se mluví o pojmech CLI (Common Language Infrastructure), což je standardizované jádro celého frameworku, a BCL (Base Class Library), což je sada základních knihoven[15].



Obrázek 2.8: Struktura .NET Framework[6]

C#

C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft. Jazyk C# je založen na programovacích jazycích C++ a Java. Tento programovací jazyk lze využít k tvorbě webových aplikací, databázových programů, formulářových aplikací ve Windows a aplikací pro mobilní zařízení.

V C# neexistuje vícenásobná dědičnost – to znamená, že každá třída může být potomkem pouze jedné třídy. Toto rozhodnutí bylo přijato, aby se předešlo komplikacím a přílišné složitosti, která je spojena s vícenásobnou dědičností. Třída ale může implementovat libovolný počet rozhraní. Neexistují žádné globální proměnné a metody, všechny musí být deklarovány uvnitř tříd. Náhradou za globální proměnné a metody jsou statické metody a proměnné veřejných tříd. C# nepotřebuje a ani neobsahuje dopřednou deklaraci – pořadí deklarace metod není důležité. Jazyk C# je case sensitive – rozlišuje mezi velkými a malými písmeny. Identifikátory „hodnota“ a „Hodnota“ tedy nejsou, na rozdíl od VB .NET, ekvivalentní [1].

WPF

Je knihovna tříd pro tvorbu grafického rozhraní. Pro vytvoření uživatelsky bohatého rozhraní využívá značkovací jazyk XAML, který umožňuje oddělit funkčnost a vzhled aplikace. Cílem WPF je sjednotit uživatelské rozhraní, 2D a 3D grafiku, vektorovou a rastrovou grafiku, animace a provázat s daty audia a videa[13].

XAML

XAML je zkratka pro Extensible Application Markup Language. Je to deklarativní jazyk založený na XML. Je určený primárně k návrhu uživatelského prostředí pro technologie WPF a Silverlight. Často se však můžeme setkat s jeho alternativním způsobem využití. Můžeme jej tedy použít na cokoliv chceme a není vázán pouze na WPF, ačkoliv je v této formě asi nejpopulárnější.

Jazyk XAML slouží k usnadnění zápisu. Vše, co lze udělat pomocí něj, tak můžeme zapsat i v jazycích C#. Když to zjednoduším, tak XAMLelem říkáme, jaké objekty se mají vytvořit a jaké vlastnosti se jim mají nastavit. Tedy například ve WPF slouží primárně k sestavení objektového stromu elementů uživatelského prostředí[16].

Model–View–ViewModel

Model–View–ViewModel je návrhový vzor pro WPF aplikace. Nabízí řešení, jak oddělit logiku aplikace od uživatelského rozhraní. Kódu je pak méně, vše je přehlednější a případné změny nejsou implementační noční můrou. MVVM odděluje data, stav aplikace a uživatelské rozhraní. Samotné WPF bylo vytvořeno tak, aby se v něm MVVM používal pohodlně.

Hlavní myšlenka MVVM je prostá – vytvořit třídu, která si drží stav aplikace. Nazývá se ViewModel. Tě se dotazuje uživatelské rozhraní, které podle ní vykresluje ovládací prvky. A naopak zadá-li uživatel do uživatelského rozhraní nějaké údaje, zpropagují se automaticky do ViewModelu. WPF je pro toto použití dobře uzpůsobeno, protože díky bindingu lze deklarativně napojit uživatelské rozhraní na ViewModel.

ViewModel je nejdůležitější třída. Poskytuje všechna data pro uživatelské rozhraní, které se nazývá View. Důležité je, že poskytuje svá data v takových datových strukturách, které

vyvolávají události při jejich změně. To umožňuje uživatelskému rozhraní nová data automaticky zobrazit hned jak se ve ViewModelu změní. ViewModel má dva základní kameny. Prvním je kolekce `ObservableCollection<T>`, která hlásí, když je přidán nebo odebrán její prvek. Druhým je rozhraní `INotifyPropertyChanged`. Popisuje událost, která nastane, když se změní některá z vlastností ViewModelu.

Model

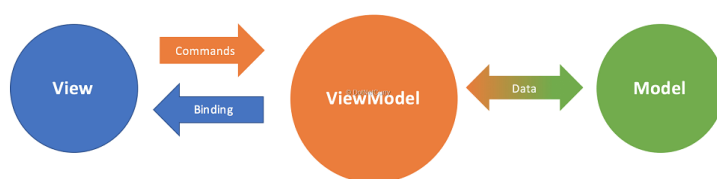
Model popisuje data, se kterými aplikace pracuje. Model nesmí o stavu ovládacích prvků nic vědět.

View

View reprezentuje uživatelské rozhraní v jazyce XAML. Může se jednat o okno aplikace, stránku, nebo ovládací prvek.

ViewModel

ViewModel spojuje Model a View a drží si stav aplikace. Ovládací prvky jsou pomocí bindingu propojeny s ViewModelem a čerpají z něj svůj obsah. Provádí se v něm filtrování dat v závislosti na stavu aplikace[14].



Obrázek 2.9: Model-View-ViewModel[11]

Reflexe

Jedním z důsledků použití mechanismu metadat pro všechny typy v prostředí .NET frameworku je možnost tyto typy v našem programu prozkoumávat a tato věc je nazývána reflexe.

Mechanismus reflexe nám tedy umožňuje procházení a manipulaci s objektovým modelem představující konkrétní aplikaci. Metadata, která jsou reflexí využívána jsou obvykle vytvářena kompilátorem při překlada aplikace[17].

Mediator

Návrhový vzor Mediator představuje zapouzdření logiky komunikace mezi několika třídami do jedné třídy Concrete Mediator. Rozhraní Mediator definuje způsob, jakým jednotlivé objekty komunikují s centrálním prvkem vzoru, kterým je Concrete Mediator[5].

Singleton

Singleton je návrhový vzor, který využijeme při řešení problému, kdy je potřeba, aby v celém programu běžela pouze jedna instance třídy. Tento návrhový vzor zabezpečí, že třída bude mít pouze jedinou instanci a poskytne k ní globální přístupový bod[18].

Kapitola 3

Analýza požadavků a návrh řešení

Tato kapitola popisuje návrh grafického uživatelského rozhraní, jsou zde rozebrány specifikace požadavků a studium cílové skupiny s příklady užití. Dále jsou zde uvedena již existující podobná řešení. Poslední částí kapitoly je návrh grafického uživatelského rozhraní a nástroje pro jeho tvorbu.

3.1 Specifikace požadavků

Cílem práce bylo vytvořit program pro zobrazování genealogických dat. Program má být schopen zobrazit všechny základní rodokmeny, které se v genealogii využívají. V rodokmenech bude možné filtrovat, což umožní jejich zobrazení pouze po zvolené generaci.

Program má umět vypočítat různé statistiky a zobrazit je ve formě grafů. Mezi tyto statistiky patří například: střední délka života, počet žijících osob, povolání a další. Statistiky by mělo být možné filtrovat podle data. Dále by mělo být možné exportovat rodokmeny do různých formátů.

3.2 Studium cílové skupiny a případy užití

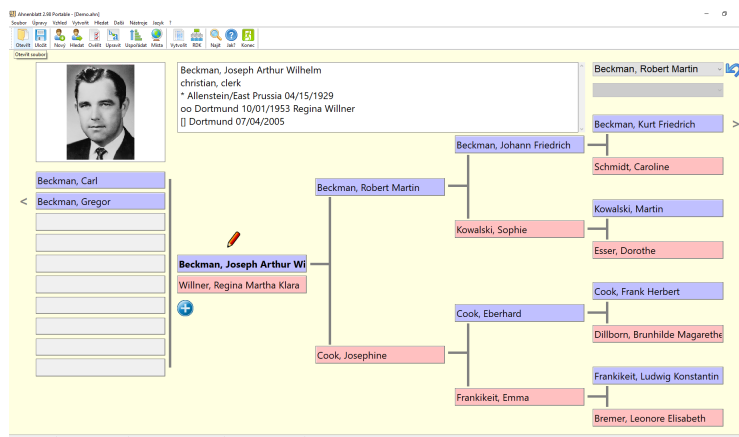
Program zobrazuje genealogická data, takže uživatel musí vlastnit již vytvořený GEDCOM soubor. Cílová skupina jsou zejména lidé, kteří se věnují genealogii, ať už na profesionální nebo amatérské úrovni. Program však mohou užívat i lidé, kteří se genealogii nevěnují a pouze si nechali vytvořit rodokmen v elektronické podobě. Program by proto měl být natolik jednoduchý, aby s ním zvládli pracovat i uživatelé, kteří se o genealogii nezajímají. Zároveň by měli zacházení s aplikací zvládnout i uživatelé se základními znalostmi práce na počítači.

Program bude využíván pro zobrazení genealogických dat, která jsou uložena v elektronické podobě. Zdrojová data budou zpracována a následně převedena do graficky přívětivé podoby. Program bude také využíván pro zobrazení statistik ve formě přehledných grafů.

3.3 Existující řešení

Ahnenblatt

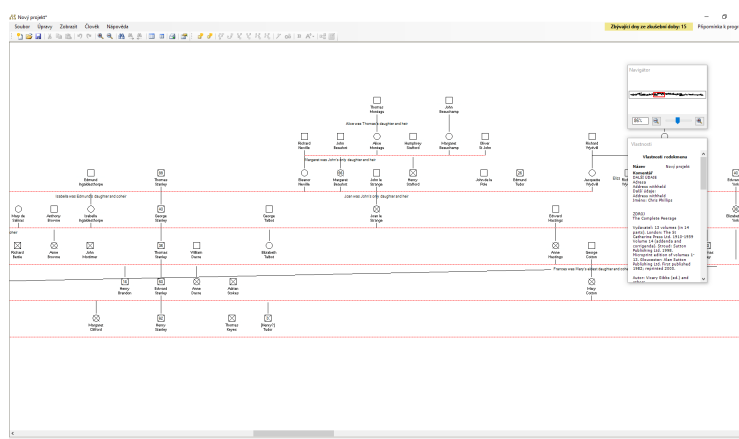
Ahnenblatt je volně dostupný genealogický program. Data ukládá ve formátu Ahnenblatt, ale dokáže importovat data z formátu GEDCOM a CSV. Program má příjemné uživatelské rozhraní. Na základní obrazovce je zobrazen nekompletní vývod osoby. Kompletní vývod je možné vykreslit v menu. Program umožňuje zobrazit statistiky ve formě koláčových grafů, ale není možné je filtrovat.



Obrázek 3.1: Ukázka programu Ahnenblatt

Rodokmen pro

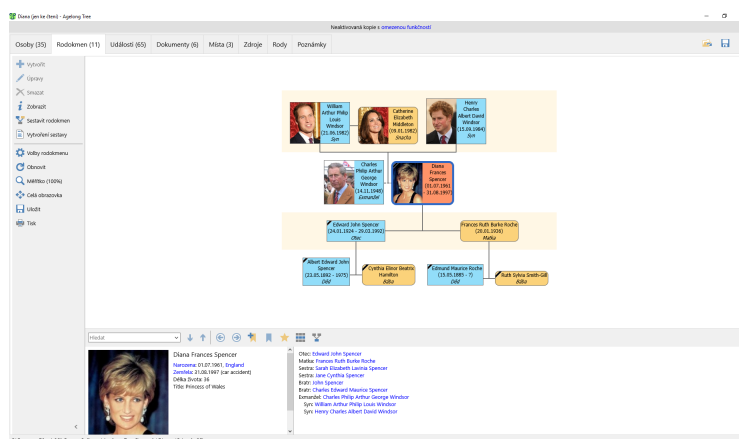
Rodokmen pro je český, placený genealogický program. Program má jednoduché uživatelské rozhraní. Na základní obrazovce je zobrazen rozrod celého rodu, bohužel poměrně nepřehledný. Program nedokáže zobrazit rodokmen konkrétní osoby. Data ukládá ve vlastním formátu Rodokmen, ale dokáže pracovat s daty ve formátu GEDCOM. Program neumožňuje zobrazit statistiky.



Obrázek 3.2: Ukázka programu Rodokmen pro

Agelong tree

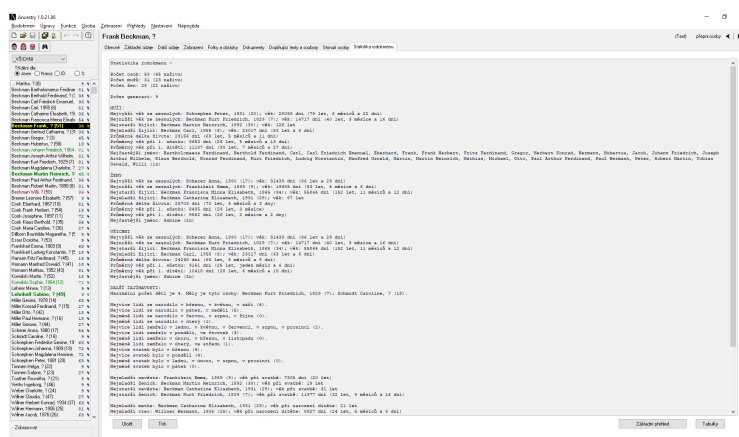
Agelong tree je placený genealogický program. Program má graficky pěkné, ale ne příliš přívětivé uživatelské rozhraní. Na základní obrazovce je zobrazen seznam osob. Vykreslení rodokmenu je možné v menu. Zobrazený rodokmen není příliš přehledný. Data ukládá ve vlastním formátu Agelong Tree, ale dokáže exportovat do formátu GEDCOM. Import dat ve formátu GEDCOM je pouze v placené verzi. Program umožňuje zobrazit jednoduché statistiky, ve kterých se nedá filtrovat.



Obrázek 3.3: Ukázka programu Agelong Tree

Ancestry

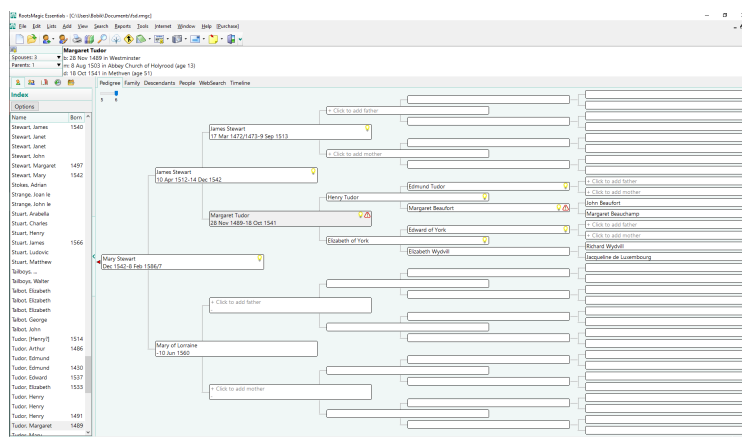
Ancestry je český, bezplatný genealogický program. Program má přívětivé uživatelské rozhraní s nemoderním vzhledem. Na základní obrazovce je zobrazen detail osoby. Vykreslení rodokmenů je možné v menu. Data ukládá ve formátu Ancestry, ale dokáže pracovat s daty ve formátu GEDCOM. Program zobrazuje rozsáhlé statistiky v textové podobě.



Obrázek 3.4: Ukázka programu Ancestry

RootsMagic Essentials

RootsMagic Essentials je bezplatný genealogický program, s příjemným vzhledem a přehledným uživatelským rozhraním. Na základní obrazovce je zobrazen seznam osob a vývod programem vybrané osoby. Vykreslený vývod není zobrazen kompletní. O vybrané osobě jsou zobrazeny pouze základní informace. Data ukládá ve formátu RootsMagic, ale dokáže pracovat s daty ve formátu GEDCOM. Program dokáže zobrazit statistiky, v bezplatné verzi programu však pouze v omezeném množství.



Obrázek 3.5: Ukázka programu RootsMagic Essentials

3.4 Návrh uživatelského rozhraní

Jednou z hlavních funkcí programu je zobrazení genealogických dat ve formě rodokmenů, proto je kladen důraz na vhodné umístění rozsáhlých genealogických tabulek na obrazovce. Další důležitou částí aplikace je zobrazení a filtrování statistik. Pro snadné a jednoduché filtrování bude vždy nad statistikami přítomen formulář pro filtrování. V okně aplikace bude vždy zobrazeno textové pole, které uživateli umožní vyhledávání mezi osobami.

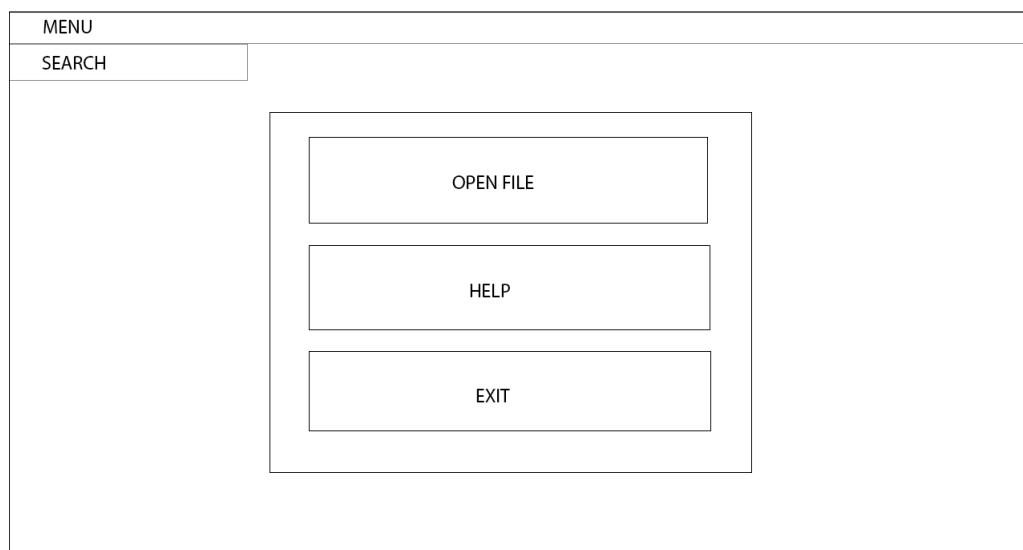
Uživatel bude moci:

- exportovat rodokmeny a grafy jako obrázek nebo dokument
- filtrovat zobrazený rodokmen
- vyhledávat osoby
- filtrovat statistiky podle zadaných kritérií

Uživateli budou zobrazeny:

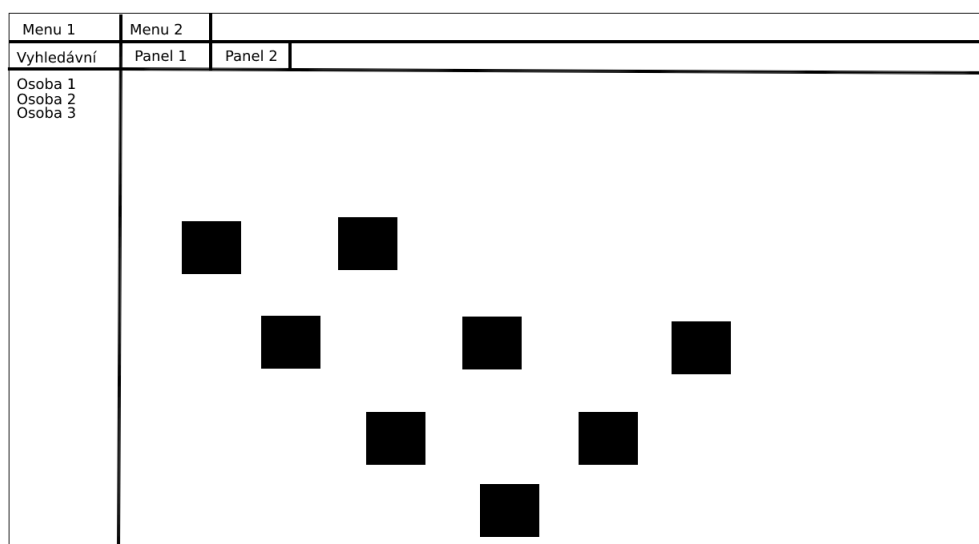
- rodokmeny
- osoby
- statistiky

Jednou z hlavních částí uživatelského rozhraní je startovací obrazovka, která musí být jednoduchá a přehledná, aby se uživatel co nejrychleji zorientoval. Z tohoto důvodu budou na startovací obrazovce pouze tři jednoduchá tlačítka. Funkcionalita těchto tlačítek bude dostupná i z hlavního menu.



Obrázek 3.6: Návrh uživatelského rozhraní – startovací obrazovka

Další důležitou částí je zobrazení rodokmenů a statistik. Protože rodokmeny i statistiky zaberou většinu místa hlavního okna, bude na hlavním okně možno přepínat mezi záložkami, kde na první záložce bude zobrazen rodokmen a na druhé budou zobrazeny statistiky. Pro snadné přepínání mezi osobami bude v okně vždy zobrazen postranní panel, ve kterém bude seznam všech osob. V panelu také bude textové pole pro vyhledávání.



Obrázek 3.7: Návrh uživatelského rozhraní – zobrazení rodokmenu

3.5 Nástroje pro tvorbu GUI

Tento program bude implementován jako desktopová aplikace a při implementaci bude využit objektově orientovaný návrh. Program bude implementován v jazyce C#, který umožňuje využití tohoto návrhu.

Grafické uživatelské rozhraní bude implementováno pomocí technologie Windows Presentation Foundation, která umožňuje vytvořit bohaté uživatelské rozhraní pomocí značkovacího jazyka XAML. Díky této technologii můžeme oddělit funkčnost a vzhled aplikace, což je pro naši aplikaci vhodné.

Pro vylepšení vzhledu aplikace bude použita knihovna Material Design In XAML Toolkit, která změní podobu většiny ovládacích prvků do moderního stylu Material Design.

Kapitola 4

Realizace

Tato kapitola se zabývá realizací aplikace. Aplikace byla implementována podle návrhového vzoru MVVM (Model–View–ViewModel), který je detailněji popsán v kapitole 2.3.

4.1 ViewModely

Jak již bylo řečeno výše, uvedený program byl implementován podle návrhového vzoru MVVM, který odděluje vzhled a funkčnost aplikace. Jednou z jeho nejdůležitějších částí je ViewModel. Zde budou uvedeny použité ViewModely a krátký popis jejich funkce.

- **MainViewModel** - Hlavní viewModel, drží kontext hlavního okna. Stará se o obsloužení událostí z menu, vykreslení rodokmenů a statistik.
- **ListViewModel** - Výběr výchozí osoby pro vykreslení rodokmenu a vyhledávání mezi osobami.
- **PersonDetailViewModel** - Zobrazuje detailní informace o vybrané osobě.
- **SettingsViewModel** - Zobrazuje nastavení aplikace a obsluhuje události z okna nastavení.

Pro komunikaci mezi ViewModely se používá třída **Messenger**. Třída Messenger je implementována podle návrhového vzoru Mediator, který je blíže popsán v kapitole 2.3.

4.2 Zpracování dat

Zpracování dat je jednou z nejdůležitějších částí aplikace. Pokud by byla data špatně zpracována, zobrazené rodokmeny a statistiky by byly chybné. Před zpracováním dat bylo nejprve nutné nastudovat strukturu formátu GEDCOM. Ten obsahuje obrovské množství záznamů. Zjednodušený popis jeho struktury je uveden v kapitole 2.3.

O zpracování dat zdrojového souboru se stará třída **Parser** a metoda **Parse**, která čte zdrojová data řádek po řádku, dokud nenarazí na řádek se znakem **TRLR**, značící konec souboru. Z řádků vytváří ucelené bloky, které charakterizují jednu entitu např. osobu, rodinu. Ucelený blok je reprezentován jako seznam řetězců. Pro každý takový blok existuje v aplikaci třída, která se stará o jeho zpracování. Tyto třídy pracují na stejném principu. Na vstupu obdrží seznam řetězců ze zdrojového souboru. Seznam je poté procházen po jednotlivých položkách a data ukládá do příslušného modelu. Pokud narazí na záznam, u kterého

se zvyšuje úroveň zanoření, vytvoří seznam řetězců a zavolá třídu, která se stará o zpracování záznamu tohoto typu.

Aplikace pracuje pouze se záznamy typu osoba a rodina. Tyto záznamy jsou nutné pro správný chod aplikace. Zpracování ostatních typů záznamů by pracovalo na stejném principu.

Pro každý typ záznamu v aplikaci existuje odpovídající model. Výsledkem zpracování je seznam osob a rodin.

4.3 Vykreslování rodokmenů

Zobrazení osoby

Každá osoba je v rodokmenu zobrazena jako instance třídy `FamilyNode`, která dědí od třídy `Button`. Oproti klasickému tlačítku třída `FamilyNode` obsahuje informace, které jsou důležité při vytváření rodokmenů. Jedná se například o osobu, která je s tímto objektem svázána. Po kliknutí na tlačítko uvnitř rodokmenu se rodokmen překreslí se změněnou výchozí osobou.

Vazby mezi osobami

Vazby mezi osobami jsou vykresleny pomocí třídy `Line`. Každé propojení je vytvořeno pomocí tří instancí třídy `Line`, aby se zamezilo protínání čar s tlačítky a bylo dosaženo přívětivějšího tvaru, připomínajícího písmeno Z.

Zobrazení rodokmenů

Vykreslování všech typů rodokmenů funguje na stejném principu. Vstupem je vybraná osoba. Pro tuto osobu se vytvoří instance třídy `FamilyNode`, popsané výše a nastaví se mu pozice na plátně. K tomu slouží vlastnost `Margin`, ve které se nastavuje odsazení od levého okraje a horního okraje. Podle typu rodokmenu se vyberou děti nebo rodiče osoby na vstupu a pro každého z nich se znovu zavolá metoda pro vykreslení rodokmene. Algoritmus tedy pracuje rekurzivně.

Pro zobrazení rodokmenu na obrazovce je použit ovládací prvek `ItemsControl`, který umožňuje promítnout kolekci prvků do panelu, uvnitř okna aplikace. Uvnitř ovládacího prvku `ItemsControl` je definován panel, na kterém mají být prvky zobrazeny. V aplikaci byl použit panel `Canvas` neboli plátno, na který je možno vykreslit tlačítka a čáry potřebné pro zobrazení rodokmenu.

Jako zdroj dat pro ovládací prvek `ItemsControl` je použita instance třídy `CompositeCollection`, která spojuje více kolekcí různého typu do jedné kolekce. V programu je to použito pro sloučení seznamu tlačítek a čar.

O vykreslení rodokmenů se stará třída `FamilyLogic`, která obsahuje metody `GetVývod`, `GetRozrod`, `GetRodokmen`. Jejich výstupem je právě `CompositeCollection`.

Vývod

Pro vykreslení vývodu je nejprve nutné zjistit úroveň výsledného rodokmenu. Z úrovně rodokmenu zjistíme maximální počet osob na nejvyšší úrovni, pokud by byla dostupná všechna data. Z úrovně rodokmenu vypočítáme potřebnou šířku plátna. Počet osob na nejvyšší úrovni se získá podle vzorce 2^n , kde n je úroveň rodokmenu.

Výchozí osoba je umístěna do středu plátna na spodní okraj. Pozice následujícího předka, oproti výchozí osobě se vypočítá pomocí vzorce:

$$x = \pm p - \frac{p}{2}$$

kde x je výsledný posuv od předchůdce a p je poloha předchůdce.

Protože zpravidla na poslední úrovni nebude maximální počet osob, vznikaly by na stránkách dlouhá, prázdná místa. Po vykreslení rodokmenu se změní šířka plátna tak, aby odpovídala šířce rodokmenu.

Rodokmen

Vykreslení rodokmenu pracuje na stejném principu jako vykreslení vývodu. Při vykreslení rodokmenu se ale nesleduje linie matky, proto roste rodokmen jen do jedné strany.

Rozrod

Rozrod je nejrozsáhlejší typ rodokmenu. Jeho šířka se nedá vypočítat na základě jednoduchého vzorce, jako je tomu u vývodu. U vývodu byly vždy maximálně dva předci. U rozrodu to může být teoreticky jakékoliv kladné číslo. Proto se na začátku algoritmu definuje dostatečně velké plátno. Výchozí osoba se umístí doprostřed, k hornímu okraji. Na základě počtu dětí se pro prvního potomka vypočítá potřebné místo a umístí se nalevo od výchozí osoby. Následující osoby jsou umísťovány za již přidanou osobu s největším odsazením od pravého okraje.

4.4 Statistiky

U statistik velice záleží na správnosti dat ve zdrojovém GEDCOM souboru. Program pracuje se dvěma typy statistik. Prvním typem jsou rodinné statistiky, které jsou vypočítány z dat celé rodiny. Druhým typem jsou osobní statistiky, které pracují s daty pouze jedné osoby, případně s výsledkem rodinné statistiky. Pro zobrazení statistik je opět použit ovládací prvek `ItemsControl`, tentokrát ale s panelem `WrapPanel`. Ten umísťuje prvky vedle sebe do řádku, pokud je orientován horizontálně. Pokud narazí na konec řádku, začne prvky umísťovat od začátku nového řádku. Každá statistika implementuje rozhraní `IStatistika`. O získání statistik se stará třída `StatistikaInstaller`. Ta pomocí reflexe, na základě implementace rozhraní `IStatistika`, vybere všechny statistiky. Reflexe je blíže popsána v kapitole 2.3. Implementace této funkcionality je vidět na následující ukázce kódu metody `GetStatistiky` z třídy `StatistikaInstaller`.

```
public List<Type> GetStatistiky()
{
    return AppDomain.CurrentDomain.GetAssemblies()
        .SelectMany(a => a.GetTypes())
        .Where(t => typeof(IStatistika).IsAssignableFrom(t)
            && !t.IsInterface)
        .ToList();
}
```

Rozhraní `IStatistika`, definuje metody `HasData` a `GetData`. Třída `StatistikaManager` projde všechny typy vrácené metodou `GetStatistiky` a vytvoří z nich objekty. Pro každý objekt se

zavolá metoda `HasData`, která určuje zda jsou dostupná potřebná data pro výpočet statistiky. Pokud ano, zavolá se metoda `GetData`. Výstupem je seznam statistik. Pro zobrazení statistik byla použita knihovna `LiveCharts WPF`¹. V třídě `MainViewModel` je ze statistik vytvořen seznam uživatelských prvků `CustomChart`. Který kromě grafu obsahuje ještě `Label` s názvem statistiky.

Díky tomuto řešení nemusíme pro každou statistiku definovat ovládací prvek. Statistika jsou dynamicky zobrazovány na základě zdrojových dat. Pro přidání nové statistiky stačí pouze vytvořit novou třídu implementující rozhraní `IStatistika`. To ulehčuje budoucí rozšiřitelnost programu.

Osobní statistiky pracují na podobném principu. Aby se výsledek zobrazil na detailu vybrané osoby, musí být dostupná data, která jsou potřebná pro výpočet statistiky. Pokud statistika pracuje s výsledkem rodinné statistiky, tak i její výsledek musí být dostupný.

Filtrování statistik

Metoda `GetData`, má definované tři nepovinné argumenty. Konkrétně počáteční datum, konečné datum a příznak `bezDat`. Tento příznak určuje, zda se má počítat s osobami, které nemají dostatečná data pro výpočet. Například pokud vezmeme statistiku pro výpočet počtu žijících osob, mají se osoby, u kterých není známo datum narození, brát jako žijící osoba nebo ne? V programu to může rozhodnout uživatel pomocí zaškrtnutí políčka. Následuje definice metody `GetData` z rozhraní `IStatistika`.

```
object GetData(DateTime? datumOd = null, DateTime? datumDo = null,
               bool bezDat=false);
```

4.5 Nastavení

Důležitou součástí každého programu je nastavení. Je samozřejmostí, že každý uživatel preferuje jiné vizuální pojetí. Proto program umožňuje změnit vzhled rodokmenu. Pro implementaci nastavení se využívá systémová knihovna `Configuration`.

O ukládání a načítání nastavení z aplikačního konfiguračního souboru se stará třída `SettingsModel`. Pokud jsou v konfiguračním souboru uložena potřebná data, třída je načte. Jinak nejsou v konfiguračním souboru nalezena, jsou tam uložena s výchozími hodnotami.

Konstrukci třídy `SettingsModel` zajišťuje třída `SettingsFactory`. Protože nastavení je v aplikaci pouze jedno a musí být stále aktuální, je `SettingsFactory` implementováno podle návrhového vzoru `Singleton`, který je popsán v kapitole 2.3. Následuje ukázka kódu pro konstrukci nastavení.

```
private static SettingsModel settings;
private static readonly object lockObject = new object();

public static SettingsModel GetSettings()
{
    lock (lockObject)
    {
        return settings ?? (settings = new SettingsModel());
    }
}
```

¹<https://lvcharts.net>

4.6 Ukládání předchozích vybraných osob

Pro jednodušší a příjemnější práci s programem bylo implementováno tlačítko zpět, pro vrácení předchozí vybrané osoby. Při každé změně vybrané osoby, ke které může dojít v postranním panelu, na detailu osoby a v rodokmenu, se vybraná osoba uloží na vrchol zásobníku. Toto chování je implementováno ve třídě BackupFactory.

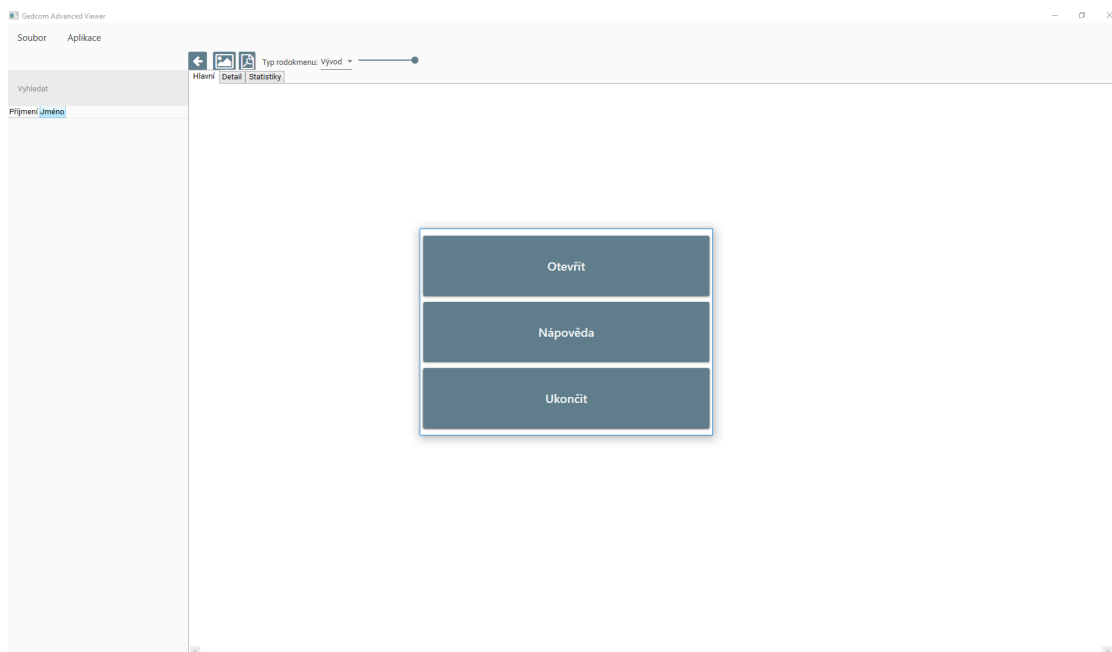
Po stisku tlačítka zpět se osoba z vrcholu zásobníku nastaví jako vybraná osoba a je vykreslen její rodokmen.

O konstrukci zásobníku se stará třída BackupFactory. Protože zásobník s předchozími osobami musí být v aplikaci pouze jeden a být stále aktuální, byla tato třída implementována podle návrhového vzoru Singleton, který je popsán v kapitole 2.3.

4.7 Výsledné uživatelské rozhraní

Konečné uživatelské rozhraní bylo vytvořeno pomocí technologie WPF a značkovacího jazyka XAML. Pro vylepšení vzhledu byla použita knihovna Material Design In XAML Toolkit², která převede vzhled ovládacích prvků do stylu Material design³.

Při vytváření finální podoby aplikace byl kladen důraz na jednoduchost a přehlednost. Proto se hned po spuštění aplikace zobrazí okno se třemi velkými tlačítky. Tlačítka nabízejí ty nejdůležitější akce potřebné po spuštění programu. Konkrétně otevření zdrojového GEDCOM souboru, zobrazení nápovědy a ukončení programu.

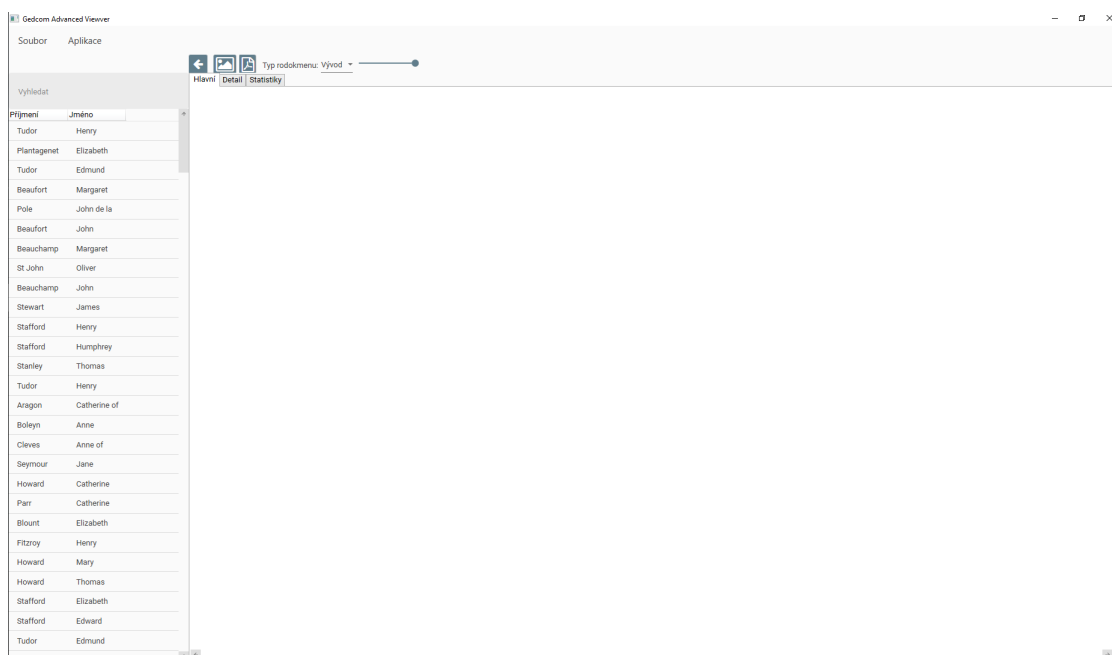


Obrázek 4.1: Ukázka startovací obrazovky

Po otevření zdrojového souboru je v postranním panelu zobrazen seznam osob a pole pro vyhledávání. Po otevření souboru probíhá výpočet statistik. Statistiku je tedy možné vidět i bez vybrané osoby na záložce Statistika.

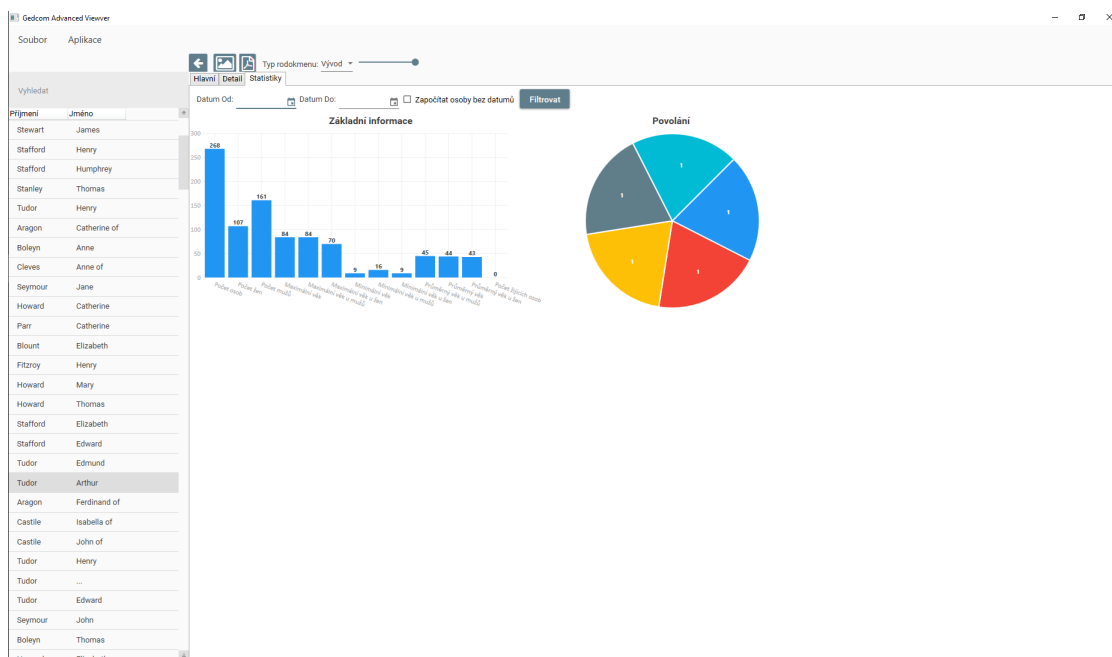
²<http://materialdesigninxaml.net>

³<https://material.io/design/>



Obrázek 4.2: Ukázka programu po otevření zdrojového souboru

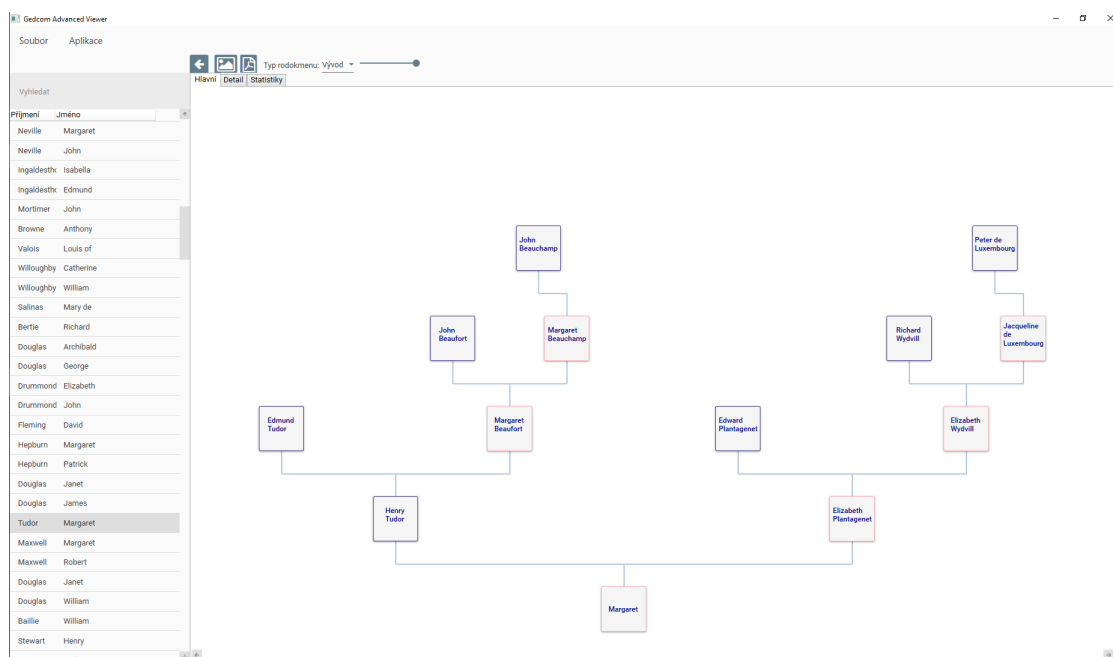
Po kliknutí na záložku statistiky se zobrazí statistiky ve formě grafů a formulář pro filtrování. Filtrování umožňuje vypočítat statistiky jen pro vybrané období.



Obrázek 4.3: Ukázka statistik

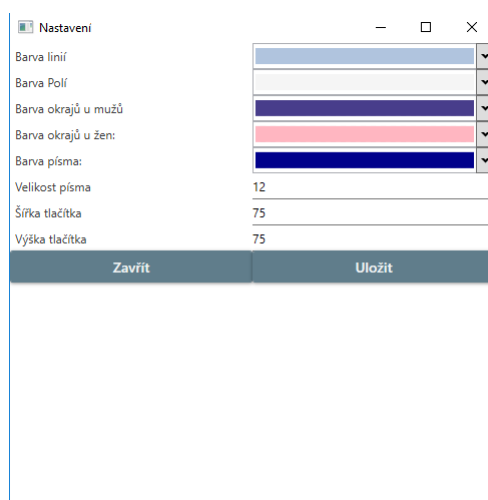
Po vybrání výchozí osoby z postranního panelu se na hlavní záložce vykreslí rodokmen. Typ rodokmenu je určen pomocí rozbalovacího menu v horní liště. Výchozím typem rodokmenu je vývod. V této liště se nachází posuvník, který umožňuje schovat úrovně rodokmenu. V liště se dále nachází tlačítka pro uložení rodokmenu nebo grafů jako obrázek

nebo dokument. Posledním tlačítkem je tlačítko zpět, které umožňuje přechod na již dříve vybranou osobu.



Obrázek 4.4: Ukázka hlavního panelu

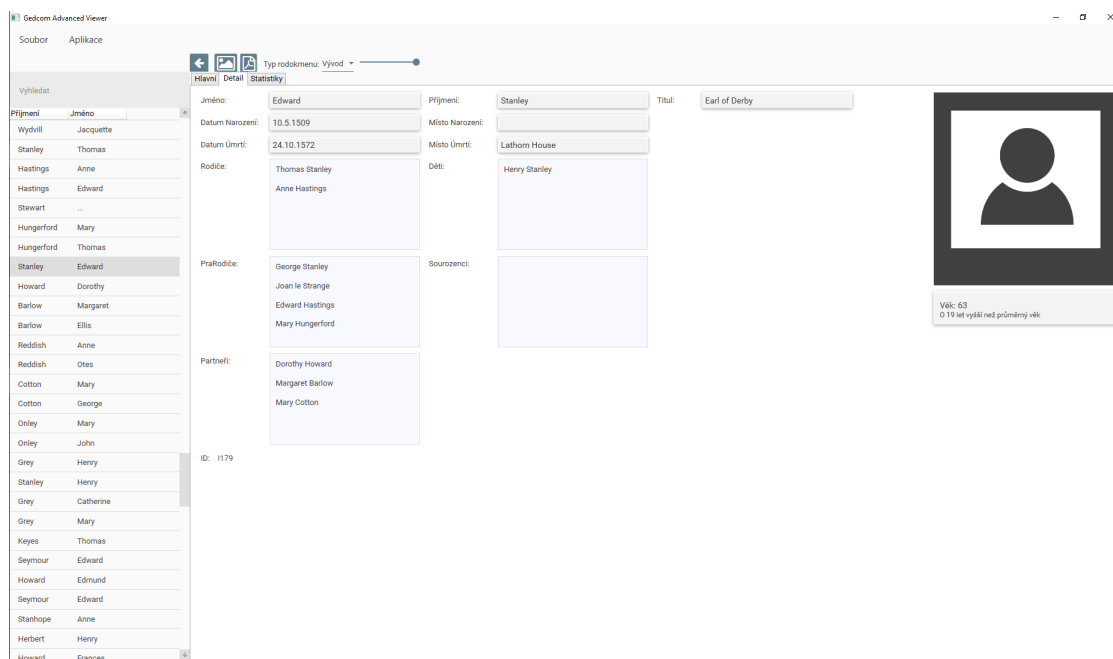
Vzhled rodokmenů je možné upravit v nastavení. Nastavení lze zobrazit v horním menu. V nastavení je možné změnit rozměry a barvy tlačítek a čar. Pro výběr barev je použit ovládací prvek `Color Picker` z knihovny `Extended WPF toolkit`⁴.



Obrázek 4.5: Ukázka nastavení

⁴<https://github.com/xceedsoftware/wpftoolkit>

Informace o vybrané osobě jsou zobrazeny na záložce Detail. Kromě základních informací se zde nachází rozříděný výpis příbuzných. Konkrétně partneři, rodiče, prarodiče, sourozenci a děti. Výpis je zobrazen pomocí ovládacího prvku `ListBox`. Po kliknutí na osobu ve výpisu se změní vybraná osoba a detail se aktualizuje. V pravé části detailu se nachází osobní statistiky. Ty jsou vykresleny v panelu `StackPanel`. Pokud osoba nemá žádné osobní statistiky, je panel skryt.



Obrázek 4.6: Ukázka panelu detail

Kapitola 5

Testování

Důležitou částí při vývoji aplikace je testování. Testování slouží k odhalení chyb a zjištění spokojenosti uživatelů se vzhledem a funkcí aplikace.

Testování proběhlo formou dotazníku. Jako dotazovací systém byl použit volně dostupný systém Survio¹. Dotazník se skládal z jednoduchých úkolů a otázek. Testování se zúčastnilo celkem 20 lidí, zejména mladšího věku, s rozličnými znalostmi práce s počítačem. Testovací formulář se skládal z otázek s jednou odpovědí a hodnotících otázek.

5.1 Struktura dotazníku

Úkoly:

1. Spustit aplikaci a otevřít zdrojový GEDCOM soubor
2. Vyhledat nebo vybrat konkrétní osobu
3. Změnit typ zobrazeného rodokmenu z vývodu na rozrod
4. Změnit nastavení aplikace
5. Zobrazit statistiky a aplikovat filtr na počáteční datum

Otázky s jednou odpovědí:

1. Spustíte program a otevřete zdrojový GEDCOM soubor (tudor.ged), který byl přiložen k programu. Jak složitý byl tento úkon?
2. Vyhledejte nebo vyberte osobu Fitzroy Henry. Jak složitý byl pro vás tento úkon?
3. Změňte typ zobrazeného rodokmenu z vývodu na rozrod. Jak složitý byl pro vás tento úkon?
4. Změňte nastavení aplikace tak, aby muži zobrazení v rodokmenu(1) měli zelený okraj. Jak složitý byl pro vás tento úkon?
5. Zobrazte statistiky rodu a vyfiltrujte od data 18.5.1500 . Jak složitý byl pro vás tento úkon?

¹<https://www.survio.com/cs/>

6. Hodnotíte program jako přehledný?
7. Použili byste program znovu?
8. Byl text a všechny prvky uživatelského rozhraní dostatečně viditelné?
9. Rušily vás nějaké grafické prvky?

Na tyto otázky bylo možné odpovědět:

1. jednoduché
2. s obtížemi, ale zvládl/a jsem to
3. obtížné

Hodnotící otázky

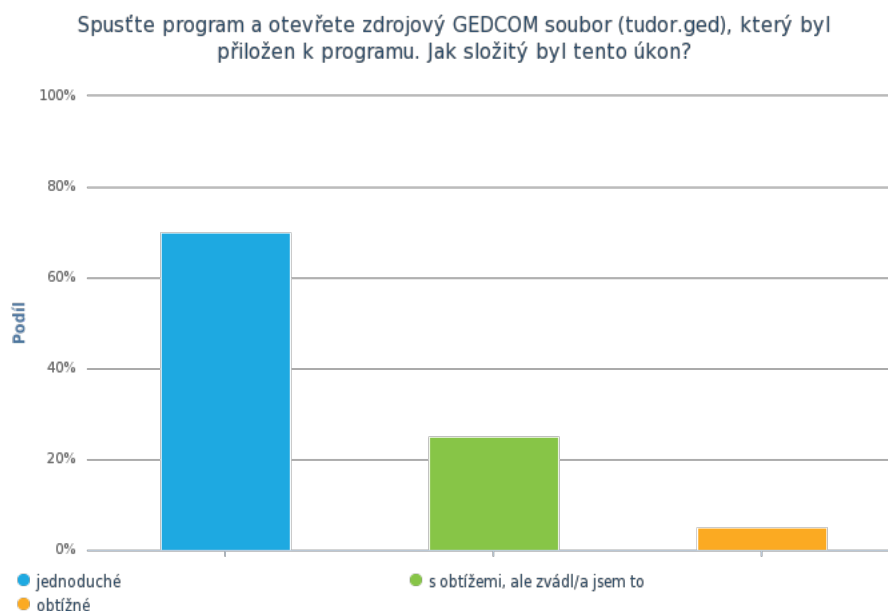
1. Uvedte, jak se vám líbí uživatelské prostředí?
2. Jak hodnotíte orientaci v programu?

Hodnotící otázky používaly systém hodnocení 1 až 5, kde 5 je nejlepší možné. V dotazníku měly odpovědi formu hvězdiček.

5.2 Odpovědi respondentů

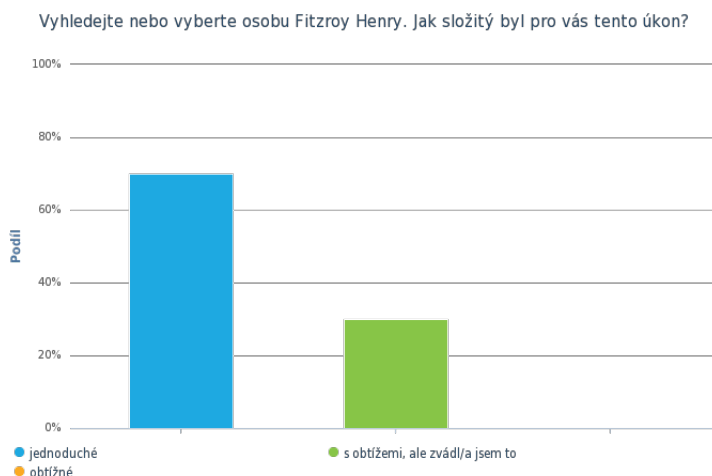
Úkoly

První úkol, který se skládal ze spuštění programu a otevření zdrojového GEDCOM souboru, zvládla většina uživatelů bez problému. Někteří uživatelé mohli mít problém s nalezením zdrojového souboru v adresářové struktuře počítače. Odpovědi na první otázku jsou znázorněny na obrázku číslo 5.1.



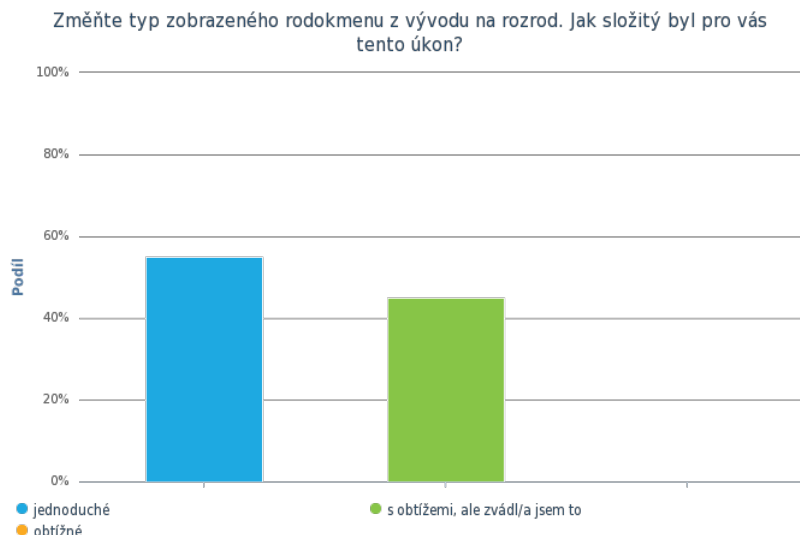
Obrázek 5.1: Graf odpovědí na úkol číslo 1

Nejjednodušším úkolem byl podle respondentů úkol číslo 2. Cílem bylo vybrat konkrétní osobu z postranního panelu, nebo ji vyhledat pomocí vyhledávacího boxu, který se také nachází v postranním panelu. Odpovědi respondentů jsou zobrazeny na obrázku číslo 5.2.



Obrázek 5.2: Graf odpovědí na úkol číslo 2

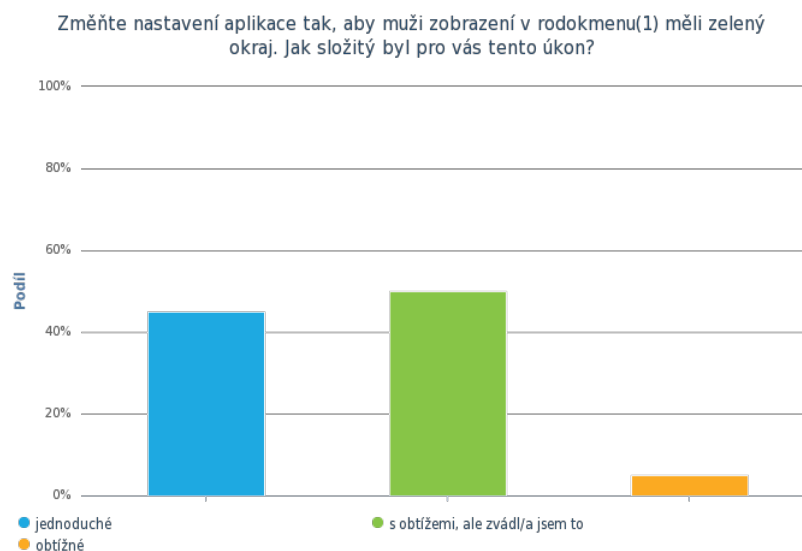
Úkol číslo 3 prověřoval horní lištu s akčními tlačítky. Úkolem bylo změnit typ zobrazeného rodokmenu. Pro změnu rodokmenu se v liště nachází rozbalovací menu. Tento úkon byl splněn většinou uživatelů bez problémů, jenom někteří měli mírné obtíže. Odpovědi respondentů jsou znázorněny na obrázku 5.3



Obrázek 5.3: Graf odpovědí na úkol číslo 3

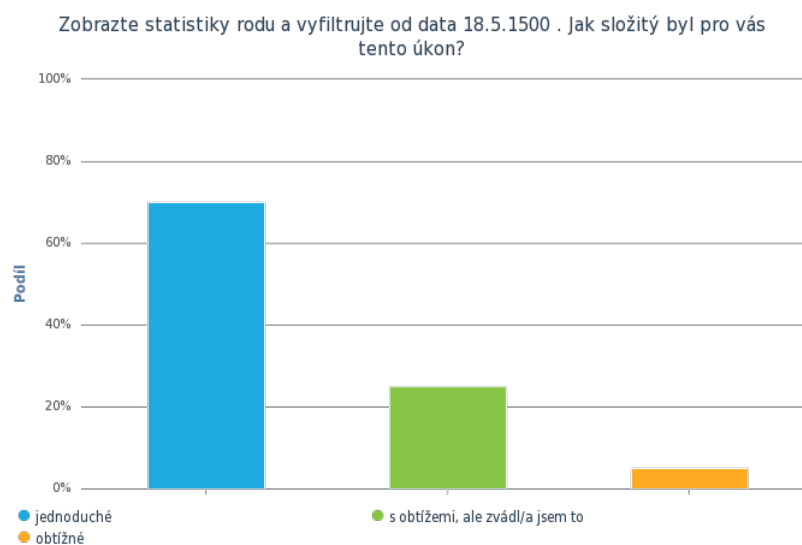
Subjektivně nejtěžším úkolem byl úkol číslo 4, což bylo promítnuto i v odpovědích respondentů. Tento úkol se skládal ze zobrazení nastavení a změny barvy okraje tlačítka. Odpovědi respondentů jsou znázorněny na obrázku číslo 5.4. Většina respondentů zvládla úkol s obtížemi. To mohlo být způsobeno tlačítkem pro přístup do nastavení, které je pouze

v hlavním menu. Řešením by mohlo být umístění tlačítka pro zobrazení nastavení do horní lišty k ostatním akčním tlačítkům.



Obrázek 5.4: Graf odpovědí na úkol číslo 4

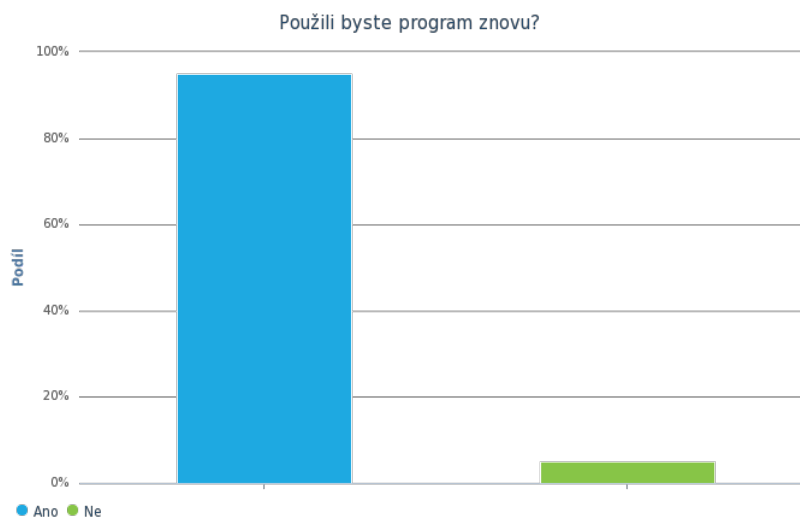
Na obrázku 5.5 jsou zobrazeny odpovědi na otázku číslo 5. Úkol se skládal z přepnutí zobrazeného panelu na panel se statistikami a filtrování dat od konkrétního data. Většina respondentů shledala úkol jednoduchým. Jeden respondent považoval úkol obtížný. To mohlo být způsobeno špatně zadaným formátem data, nebo vyhledáním tlačítka pro přepnutí z hlavního panelu na panel se statistikami.



Obrázek 5.5: Graf odpovědí na úkol číslo 5

Otázky

Respondenti na položené otázky odpovídali převážně kladně. Vzhled aplikace byl ohodnocen poměrně vysokým hodnocením. Většině respondentů přišel program přehledný. Z odpovědí také vyplývá, že všechny grafické prvky uživatelského rozhraní byly dostatečně viditelné. Téměř všichni respondenti by program použili znovu, což je zobrazeno na obrázku číslo 5.6.



Obrázek 5.6: Graf budoucího použití aplikace

Z připomínek respondentů vyplývá, že by respondenti ocenili větší možnost modifikace zobrazených rodokmenů a přidání dalších statistik, na což by bylo dobré se zaměřit při dalším vývoji aplikace.

5.3 Další vývoj

Z velmi kladných ohlasů při testování vyplývá, že další vývoj aplikace by byl více než vhodný. Při dalším vývoji aplikace by bylo vhodné se zaměřit na rozšíření nastavení a statistik, jak bylo zmíněno výše.

Dalším krokem vývoje by mohlo být rozšíření zpracování souboru. Jak bylo zmíněno v kapitole 4.2 program zpracovává pouze záznamy typu rodina a osoba. Rychlost zpracování souboru by mohla být vylepšena pomocí asynchronní implementace, protože na pořadí zpracování záznamů nezáleží.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo navrhnout a vytvořit program pro zobrazování genealogických dat. Tato aplikace umožňuje uživateli zobrazit „surová“ genealogická data v příjemné grafické podobě. Důležitou částí této aplikace je zobrazení statistik, ale hlavním přínosem je možnost filtrování podle zadaných kritérií.

Před vytvořením aplikace bylo nejprve nutné nastudovat teoretické znalosti potřebné pro vytvoření tohoto programu. Hlavní částí studia byl rozsáhlý formát GEDCOM, který bylo nezbytné nastudovat pro zpracování zdrojového souboru. Dalšími potřebnými částmi byly technologie a nástroje potřebné pro tvorbu programu.

Výsledkem této práce je program, který splňuje určené cíle. Program zpracuje soubor ve formátu GEDCOM a umožňuje zobrazit základní rodokmeny a statistiky. Statistiku je možné filtrovat. Dále také umožňuje modifikovat vzhled rodokmenů.

Výsledky testování byly převážně pozitivní. Uživatelé pozitivně hodnotili uživatelské prostředí. Většina respondentů by program použila znovu.

Z hlediska dalšího pokračování práce na tomto programu by bylo vhodné rozšířit zpracování zdrojového souboru i o vedlejší záznamy. Dále by se daly rozšířit statistiky.

Práce pro mne byla velmi přínosná, především rozšířením znalostí v oblasti tvorby grafického uživatelského rozhraní pomocí technologie WPF.

Literatura

- [1] *C Sharp*. [Online; navštíveno 14.02.2018].
URL https://cs.wikipedia.org/wiki/C_Sharp
- [2] *GEDCOM*. [Online; navštíveno 12.02.2018].
URL <https://cs.wikipedia.org/wiki/GEDCOM>
- [3] *Genealogie*. [Online; navštíveno 12.02.2018].
URL <https://cs.wikipedia.org/wiki/Genealogie>
- [4] *Grafické uživatelské rozhraní*. [Online; navštíveno 26.03.2018].
URL https://cs.wikipedia.org/wiki/Grafické_uživatelské_rozhraní
- [5] *Mediator Pattern*. [Online; navštíveno 21.03.2018].
URL https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=32240
- [6] *.Net Frameworks Architecture*. [Online; navštíveno 26.03.2018].
URL <http://zappmania.in/2013/02/25/net-frameworks-architecture.htm>
- [7] *Rodokmen*. [Online; navštíveno 12.02.2018].
URL <https://cs.wikipedia.org/wiki/Rodokmen>
- [8] *Rozrod*. [Online; navštíveno 12.02.2018].
URL <https://cs.wikipedia.org/wiki/Rozrod>
- [9] *Simple GEDCOM*. [Online; navštíveno 17.03.2018].
URL <http://heiner-eichmann.de/gedcom/simple.ged>
- [10] *Tvorba uživatelských rozhraní*. [Online; navštíveno 26.01.2017].
URL <https://www.fit.vutbr.cz/study/courses/ITU/private/lectures/Intro/itu-gui-cs.uvod.pdf>
- [11] *Using MVVM in your Xamarin.Forms app*. [Online; navštíveno 26.04.2018].
URL <http://www.dotnetcurry.com/xamarin/1382/mvvm-in-xamarin-forms>
- [12] *Vývod (genealogie)*. [Online; navštíveno 12.02.2018].
URL [https://cs.wikipedia.org/wiki/V%C3%BDvod_\(genealogie\)](https://cs.wikipedia.org/wiki/V%C3%BDvod_(genealogie))
- [13] *Windows Presentation Foundation*. [Online; navštíveno 12.02.2018].
URL https://cs.wikipedia.org/wiki/Windows_Presentation_Foundation
- [14] Dajbych, V.: *Model-View-ViewModel*. [Online; navštíveno 26.04.2018].
URL <https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>

- [15] Herceg, T.: *Úvod do .net frameworku*. [Online; navštíveno 26.03.2018].
URL <https://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku>
- [16] Jecha, T.: *Jazyk xaml*. [Online; navštíveno 26.04.2018].
URL <https://www.dotnetportal.cz/clanek/198/Jazyk-XAML>
- [17] Puš, P.: *Poznáváme C# a Microsoft.NET 36. díl – úvod do reflexe*. [Online; navštíveno 22.03.2018].
URL <https://www.zive.cz/clanky/poznavame-c-a-microsoftnet-36-dil-uvod-do-reflexe/sc-3-a-126122/default.aspx>
- [18] Žoltá, L.: *Singleton*. [Online; navštíveno 24.04.2018].
URL <http://lucie.zolta.cz/index.php/softwareve-inzenyrstvi/138-singleton>

Příloha A

Obsah CD

- bin – přeložený program
- src – zdrojový kód
- doc – dokumentace
- pdf – dokumentace ve formátu pdf
- navod.txt – návod pro překlad programu
- sample – testovací GEDCOM soubory